AI SERVICE HANDOVER & AWS DEPLOYMENT GUIDE

Prepared for: Mahmoud

Prepared by: Ayed

AI SERVICE HANDOVER - FULL TECHNICAL DOCUMENTATION + AWS GUIDE

1. Overview

This document provides a complete technical handover of the AI Service for dental clinics, covering the voice agent, analytics system, PMS integration, public API endpoints, environment configuration, runtime behavior, and AWS deployment strategy. All endpoints are public and do not require JWT or additional authentication.

2. System Components

2.1 Voice Agent

The voice agent orchestrates conversations using Gemini (via @google/genai), performing intent detection, entity extraction, and leveraging clinic-specific templates stored in Postgres.

2.2 Analytics

The analytics module captures session_started, session_closed, and turn events. Reports aggregate event counts and conversational patterns consumed by the dashboard.

2.3 PMS Integration

The integration layer communicates with PMS providers to book, update, cancel appointments, and submit performance KPIs.

3. Quick Start (Local Development)

- Create `.env.local` from `.env.example`.
- Required variables: GEMINI_API_KEY, GEMINI_TEXT_MODEL (optional), PROJECT_ID, DATABASE_URL.
- Commands: npm install; npx prisma migrate deploy; npm run seed (optional); npm start.
- Local environment runs on https://localhost:3000.

4. Public API Endpoints

Voice Agent:
- GET /api/agent/config

Analytics:
- POST /api/analytics/events
- GET /api/analytics/report

PMS:
- GET /api/integrations/pms/providers
- POST /api/integrations/pms/:provider/book
- PATCH /api/integrations/pms/:provider/booking/:bookingId
- DELETE /api/integrations/pms/:provider/booking/:bookingId
- POST /api/integrations/pms/:provider/performance

5. Error Format

All errors follow `{ "status": "error", "message": "...", "details": "..." }`.

6. Internal Structure

- Conversation Manager: src/services/conversation_manager.ts
- Database Helpers: server/dbBookingIntegration.ts
- PMS Adapters: server/pmsIntegration.ts
- Metrics & Logging: server/systemMetrics.ts, server/audit-logger.ts

7. Postman Testing

The Postman collection (`docs/AI-service.postman_collection.json`) includes analytics and PMS endpoints with no authentication headers required.

8. Updated Happy Path

1. Send analytics event.

2. Create PMS booking.

3. Fetch analytics report.

AWS DEPLOYMENT GUIDE
---

9. AWS Architecture Overview

Recommended architecture components:
- AWS ECS Fargate or EC2 for hosting the Node.js service.
- AWS RDS (Postgres) for the database.
- AWS API Gateway or Application Load Balancer (ALB) for routing.
- AWS VPC with private subnets.
- AWS Secrets Manager to store API keys and DATABASE_URL.
- Amazon CloudWatch for logs and metrics.

10. Deployment Steps

1. Build Docker image for the AI Service.

2. Push the image to Amazon ECR.

3. Create an ECS task definition (Fargate recommended).

4. Inject environment variables via Secrets Manager.

5. Deploy the service to an ECS Service in a private subnet.

6. Attach an ALB or API Gateway to expose endpoints publicly.

7. Configure autoscaling rules based on CPU and memory usage.

11. Environment Configuration for AWS

Use AWS Secrets Manager for GEMINI_API_KEY, GEMINI_TEXT_MODEL, PROJECT_ID, and DATABASE_URL. Expose them to ECS using task definition environment secret mappings.

12. Logging & Monitoring

- Use CloudWatch Logs for application logs.
- Enable CloudWatch alarms for 5xx error spikes, latency increases, and ECS task restarts.

13. Scaling Considerations

The service is stateless and horizontally scalable. Configure ECS Fargate autoscaling based on CPU or memory utilization.

14. Security Notes

- Enforce API Gateway or ALB security group IP allowlists.
- Use VPC private access paths for PMS systems.
- Restrict ALB security groups to trusted networks.

15. Rollout Checklist

- Deploy the service to ECS.
- Connect AWS RDS Postgres.
- Import `.env` values into Secrets Manager.
- Validate endpoints using the Postman collection.
- Execute the happy-path flow (analytics -> PMS -> report).
- Enable monitoring and alarms.